

**Amendments to the Claims:**

The listing of claims will replace all prior versions, and listings, of claims in the application:

**Listing of Claims:**

1. (Currently Amended): A computerized method of managing a file system for a file server, comprising ~~the steps of:~~

~~maintaining a number of unallocated blocks reserved for files of the file system;~~

receiving a file operation that signals a reservation operation for a file of the file system,  
the file having a file size;

computing a first number of blocks needed to accommodate ~~write~~ the file size;

subtracting from the first number of blocks ~~needed to write the file~~ a second number of  
blocks already allocated for the file ~~to obtain a third number of blocks; subtracting from the third~~  
~~number of blocks~~ and a third number of delayed allocated blocks for the file to obtain a fourth  
number of unallocated blocks to be reserved to accommodate the file size; and

~~adjusting the number of reserved unallocated blocks by the fourth number.~~

2. (Original): A method as in claim 1, wherein the file system uses a write anywhere file system layout.

3. (Original): A method as in claim 1, wherein the file operation that signals the reservation operation is a zero length write request.

4. (Original): A method as in claim 1, wherein the file operation that signals the reservation operation includes a parameter that specifies the file size.

5. (Currently Amended): A method as in claim 1, wherein ~~the step of~~ computing comprises:  
determining a total number of direct and indirect blocks needed to accommodate the file size.

6. (Previously Presented): A method as in claim 1, further comprising:  
setting a flag in an inode for the file that indicates blocks have been reserved for the file.

7. (Currently Amended): A method according to claim 1, further comprising ~~the step of~~  
checking that a number of available blocks in the file system is greater than the fourth number of blocks, wherein an error is returned in a case that the number of available blocks is less than the fourth number of blocks.

8. (Original): A method as in claim 7, wherein the number of available blocks in the file system is determined by subtracting a number of allocated blocks, a number of cached unallocated blocks, and a number of reserved blocks from a total number of blocks in the file system, and adding a number of reserved cached unallocated blocks.

9. (Currently Amended): A method according to claim 1, further comprising ~~the step of~~  
checking that a fifth ~~the third~~ number of blocks does not exceed a remainder of a quota for an owner of the file, wherein an error is returned in a case that the fifth ~~third~~ number of blocks

exceeds the remainder of the quota, wherein the fifth number of blocks comprises a difference between the first number of blocks and the second number of blocks.

10. (Currently Amended): A method as in claim 1, further comprising ~~the step of~~ releasing reservation of blocks as blocks are written to storage.

11. (Currently Amended): A method as in claim 10, wherein ~~the step of~~ releasing reservation of blocks further comprises ~~the step of~~ decrementing the number of reserved unallocated blocks by a number of released blocks.

12-21. (Canceled)

22. (Previously Presented): A method according to claim 1, further comprising:  
caching one or more blocks of the file in a buffer;  
writing the one or more blocks to storage; and  
decrementing the number of unallocated blocks by the number of blocks written to the storage.

23. (Previously Presented): A method according to claim 22, further comprising setting a caching flag for each block cached in the buffer.

24. (Currently Amended): A file server comprising a memory storing a computer program, a processor capable of executing the program, and a storage device capable of storing files of a file system under control of the processor, wherein the program comprises:

instructions causing the processor to ~~maintain a number of unallocated blocks reserved for files of the file system;~~

instructions causing the processor to receive a file operation that signals a reservation operation for a file of the file system, the file having a file size;

instructions causing the processor to compute a first number of blocks needed to accommodate ~~write~~ the file size;

instructions causing the processor to subtract from the first number of blocks ~~needed to write the file~~ a second number of blocks already allocated for the file ~~to obtain a third number of blocks;~~ ~~instructions causing the processor to subtract from the third number of blocks~~ and a third number of delayed allocated blocks for the file to obtain a fourth number of unallocated blocks to be reserved to accommodate the file size; ~~and~~

~~instructions causing the processor to adjust the number of reserved unallocated blocks by the fourth number.~~

25. (Previously Presented): A file server according to claim 24, wherein the file system uses a write anywhere file system layout.

26. (Previously Presented): A file server according to claim 24, wherein the file operation that signals the reservation operation is a zero length write request.

27. (Previously Presented): A file server according to claim 24, wherein the file operation that signals the reservation operation includes a parameter that specifies the file size.

28. (Previously Presented): A file server according to claim 24, wherein the instructions that cause the processor to compute comprise instructions to cause the processor to determine a total number of direct and indirect blocks needed to accommodate the file size.

29. (Previously Presented): A file server according to claim 24, wherein the program further comprises instructions causing the processor to set a flag in an inode for the file, the flag indicating that blocks have been reserved for the file.

30. (Previously Presented): A file server according to claim 24, wherein the program further comprises instructions that cause the processor to check whether a number of available blocks in the file system is greater than the fourth number of blocks, and return an error in a case that the number of available blocks is less than the fourth number of blocks.

31. (Previously Presented): A file server according to claim 30, wherein the processor determines the number of available blocks in the file system by subtracting a number of allocated blocks, a number of cached unallocated blocks, and a number of reserved blocks from a total number of blocks in the file system, and adding a number of reserved cached unallocated blocks.

32. (Currently Amended): A file server according to claim 24, wherein the program further comprises instructions that cause the processor to check whether a fifth ~~the third~~ number of

blocks does not exceed a remainder of a quota for an owner of the file, and return an error if the fifth ~~third~~ number of blocks exceeds the remainder of the quota, wherein the fifth number of blocks comprises a difference between the first number of blocks and the second number of blocks.

33. (Previously Presented): A file server according to claim 24, wherein the program further comprises instructions that cause the processor to release reservation of blocks as blocks are written to storage.

34. (Previously Presented): A file server according to claim 33, wherein the instructions that cause the processor to release comprise instructions that cause the processor to decrement the number of the reserved unallocated blocks by a number of released blocks.

35. (Currently Amended): An article of manufacture comprising a memory storing a computer program, the memory being readable by a processor capable of executing the program, the processor being capable of storing files of a file system in mass storage device under control of the program, wherein the program comprises:

~~instructions causing the processor to maintain a number of unallocated blocks reserved for a plurality of files of the file system;~~

instructions causing the processor to receive a file operation that signals a reservation operation for a file of the file system, the file having a file size;

instructions causing the processor to compute a first number of blocks needed to accommodate write the file size;

instructions causing the processor to subtract from the first number of blocks ~~needed to~~  
~~write the file~~ a second number of blocks already allocated for the file ~~to obtain a third number of~~  
~~blocks; instructions causing the processor to subtract from the third number of blocks and a third~~  
number of delayed allocated blocks for the file to obtain a fourth number of unallocated blocks to  
be reserved to accommodate the file size; and

~~instructions causing the processor to adjust the number of reserved unallocated blocks by~~  
~~the fourth number.~~

36. (Previously Presented): An article of manufacture according to claim 35, wherein the file system uses a write anywhere file system layout.

37. (Previously Presented): An article of manufacture according to claim 35, wherein the file operation that signals the reservation operation is a zero length write request.

38. (Previously Presented): An article of manufacture according to claim 35, wherein the file operation that signals the reservation operation includes a parameter that specifies the file size.

39. (Previously Presented): An article of manufacture according to claim 35, wherein the instructions that cause the processor to compute comprise instructions to cause the processor to determine a total number of direct and indirect blocks needed to accommodate the file size.

40. (Previously Presented): An article of manufacture according to claim 35, wherein the program further comprises instructions causing the processor to set a flag in an inode for the file, the flag indicating that blocks have been reserved for the file.

41. (Previously Presented): An article of manufacture according to claim 35, wherein the program further comprises instructions that cause the processor to check whether a number of available blocks in the file system is greater than the fourth number of blocks, and return an error in a case that the number of available blocks is less than the fourth number of blocks.

42. (Previously Presented): An article of manufacture according to claim 41, wherein the processor determines the number of available blocks in the file system by subtracting a number of allocated blocks, a number of cached unallocated blocks, and a number of reserved blocks from a total number of blocks in the file system, and adding a number of reserved cached unallocated blocks.

43. (Currently Amended): An article of manufacture according to claim 35, wherein the program further comprises instructions that cause the processor to check whether a fifth ~~the third~~ number of blocks does not exceed a remainder of a quota for an owner of the file, and return an error if the fifth ~~third~~ number of blocks exceeds the remainder of the quota, wherein the fifth number of blocks comprises a difference between the first number of blocks and the second number of blocks.



44. (Previously Presented): An article of manufacture according to claim 35, wherein the program further comprises instructions that cause the processor to release reservation of blocks as blocks are written to storage.

45. (Previously Presented): An article of manufacture according to claim 44, wherein the instructions that cause the processor to release comprise instructions that cause the processor to decrement the number of reserved unallocated blocks by a number of released blocks.